CLAIMS

What is claimed is:

1. Memory apparatus comprising:

a byte-bank organized in N rows and 8 columns, having a capacity of $\log_2(N)$ bytes;

a $\log_2(N)$ bit address bus operative to address said byte-bank;

an address offset bus operative to generate address offsets to bits of said byte-bank with an address conversion operator; and

an adder in operative communication with said address offset bus and said $\log_2(N)$ bit address bus, said adder operative to add addresses of said byte-bank with the address offset generated by said address conversion operator and output a result to said $\log_2(N)$ bit address.

2. The memory apparatus according to claim 1, wherein said address offsets comprise one-bit address offsets.

3. The memory apparatus according to claim 1, further comprising a random access memory array comprising a plurality of said byte-banks.

4. The memory apparatus according to claim 3, wherein for each byte-bank, said address conversion operator is operative to convert an unaligned address on a byte resolution into a $\log_2(N)$ address on a data word resolution.

5. The memory apparatus according to claim 4, wherein said address offsets comprise one-bit address offsets.

6. The memory apparatus according to claim 1, wherein said address offset comprises shifting the most significant byte from the least significant byte by a distance of 8 bytes.

7. A method comprising:

providing a random access memory array comprising a plurality of said byte-banks, wherein each byte-bank is organized in N rows and 8 columns, each byte-bank having a capacity of $\log_2(N)$ bytes;

providing a $\log_2(N)$ bit address bus operative to address said byte-bank;

generating address offsets to bits of said byte-bank with an address conversion operator; and

adding addresses of said byte-bank with the generated address offset and outputting a result to said $\log_2(N)$ bit address.

8. The method according to claim 7, wherein generating the address offsets comprises generating one-bit address offsets.

9. The method according to claim 7, further comprising, for each byte-bank, converting an unaligned address on a byte resolution into a $\log_2(N)$ address on a data word resolution.

10. The method according to claim 7, wherein generating the address offsets comprises shifting the most significant byte from the least significant byte by a distance of 8 bytes.

11. The method according to claim 7, further comprising performing an unaligned memory access to said byte-banks.

12. The method according to claim 11, wherein the unaligned memory access comprises transforming a data word before a memory write operation and compensating for the data word rotation with the address offsets.

13. The method according to claim 11, wherein the unaligned memory access comprises transforming a data word before a memory read operation and compensating for the data word rotation with the address offsets.